GOVERNMENT OF TAMILNADU

DIRECTORATE OF TECHNICAL EDUCATION,
CHENNAI

NAAN MUDHALVAN SCHEME (TNSDC) SPONSORED

STUDENTS DEVELOPMENT PROGRAMME

ON

IoT AND ITS APPLICATIONS

“PERSON COUNT DETECTION CHECK-IN”

HOST INSTITUTION

XXXX

COIMBATORE - 04

TRAINING PARTNER

ENTHU TECHNOLOGY SOLUTIONS INDIA PVT LTD

DATE:

NAME	ROLL NO
Name 1	
Name 2	
Name 3	
Name 4	
Name 5	
Name 6	

Table of contents

PERSON COUNT DETECTION CHECK-IN

ABSTRACT:

In an era where efficient space management and safety monitoring are increasingly important, automated systems offer significant advantages over traditional manual methods. The "Person Count Detection Check-In" project introduces a novel approach to tracking and managing person count within a specific area using a combination of an ESP32 microcontroller, a PIR (Passive Infrared) sensor, and a 16x2 LCD display. The system operates by deploying the PIR sensor to detect motion, which is then processed by the ESP32 microcontroller to count the number of individuals entering or exiting the monitored space. The real-time count is displayed on the 16x2 LCD, providing immediate and accurate information about the number of people present. This setup not only streamlines the process of tracking attendance but also enhances the ability to manage space utilization effectively. By integrating these components, the project offers a cost-effective and user-friendly solution suitable for various applications, including office environments, public facilities, and event management. The "Person Count Detection Check-In" system demonstrates the potential of combining simple electronic components to achieve practical and valuable outcomes in person count monitoring.

INTRODUCTION:

In modern environments, such as offices, hospitals, and public facilities, managing and monitoring the flow of people is crucial for efficient operations and ensuring safety. Traditional methods of tracking attendance and checking-in can be cumbersome and prone to errors. To address these challenges, we present the "Person Count Detection Check-In" system, a mini project designed to automate and simplify the process of monitoring and recording the presence of individuals in a given area. The system employs an ESP32 microcontroller, a PIR (Passive Infrared) sensor, and a 16x2 LCD to create an effective and user-friendly solution. The ESP32, known for its versatility and connectivity, serves as the central processing unit of the system, managing the data collected by the PIR sensor and controlling the LCD display. The PIR sensor is used to detect motion and count the number of individuals entering or exiting the monitored area. The 16x2 LCD provides a clear and immediate display of the person count, offering real-time feedback. This project aims to provide a reliable and straightforward method for tracking the number of people in a specific area, which can be valuable for various applications, including monitoring room occupancy, managing resources, and ensuring safety regulations. By integrating these components, the "Person Count Detection Check-In" system offers an innovative approach to handling person count data efficiently.

HARDWARE & SOFTWARE COMPONENTS USED:

ESP32 Microcontroller

WOKWI

PIR sensor

USB Cable (B-Type)

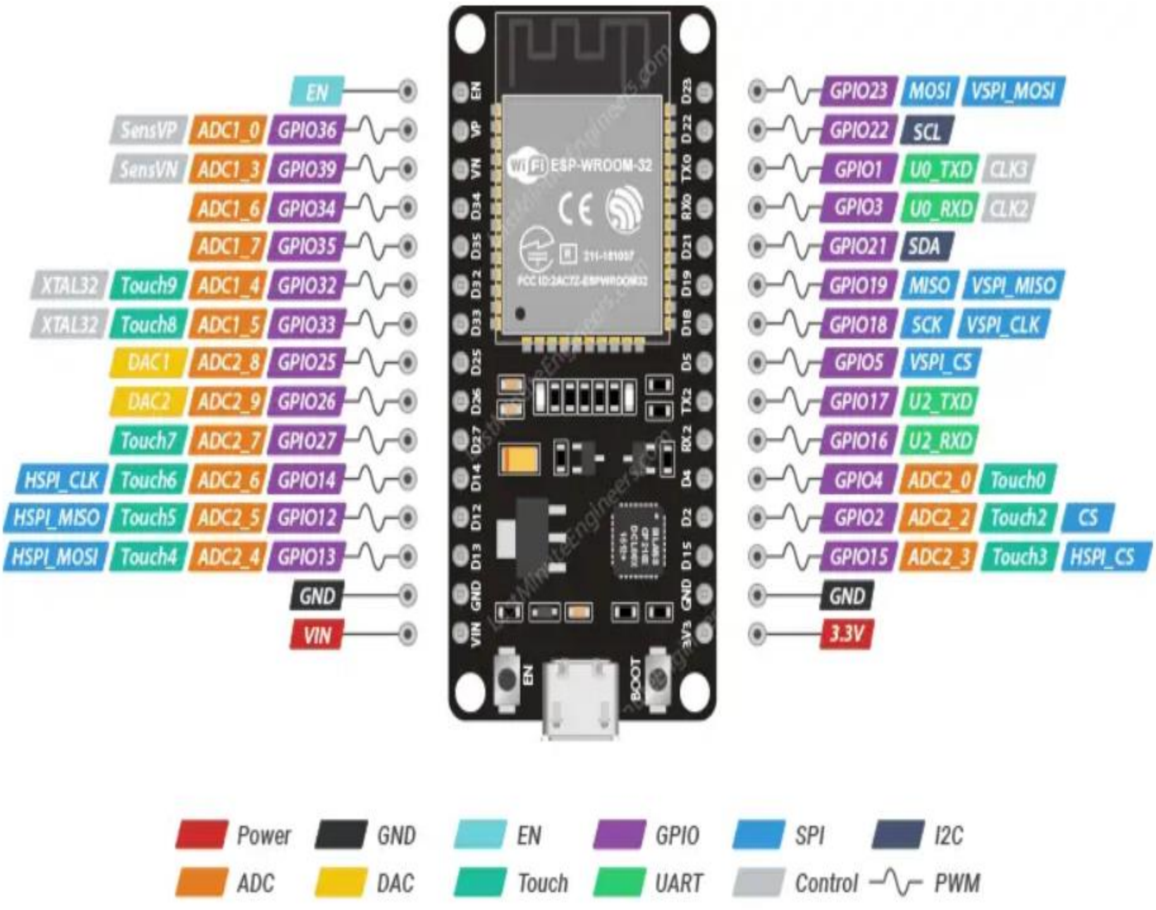
ARDUINO IDE

LCD (16x2)

HARDWARE COMPONENTS DESCRIPTION:

i) ESP32:

The ESP32 is a highly versatile microcontroller developed by Espressif Systems, designed for a wide range of applications, particularly in the Internet of Things (IoT) space. It is renowned for its combination of high performance, integrated wireless connectivity, and a rich set of features, all at a low cost. The ESP32 is commonly used in projects that require both Wi-Fi and Bluetooth capabilities, making it suitable for smart home devices, sensor networks, and wearable technology.



Key Features:

- **Dual-Core Processor**: Features a dual-core Tensilica LX6 microprocessor running up to 240 MHz.

- **Connectivity**: Includes Wi-Fi (802.11 b/g/n) and Bluetooth (Classic and BLE).
- **Memory**: Typically comes with 520 KB of SRAM and supports external flash memory.
- **I/O Pins**: Offers numerous GPIO (General Purpose Input/Output) pins with various functionalities.
- **Peripherals**: Includes ADC, DAC, PWM, SPI, I2C, UART, and more.
- **Power Management**: Equipped with low-power modes for energy efficiency.

ii) PIR SENSOR:

All living objects, whose body temperature is more than 0°C, emit the heat in form of infrared radiation through their body, also called as thermal radiations. This Radiated energy is invisible to human eye. These Signals can be detected by using PIR sensor which is specially designed for such purpose. PIR sensor i.e. Passive Infrared Sensor, passive word indicates PIR Sensor does not generate or radiate any energy for detection purposes. PIR Sensors don't detect or measure "HEAT"; they detect the infrared radiation emitted or reflected from objects. They are small, inexpensive, low power and easy to use. They are commonly found at home, medical, factories etc. areas.

Key specifications:

Detection range: up to 7 meters

Detection angle: 110 degrees

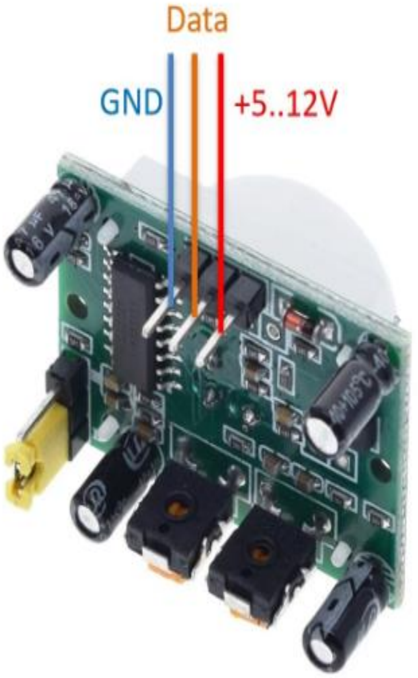
Operating voltage: DC 4.5V - 12V DC

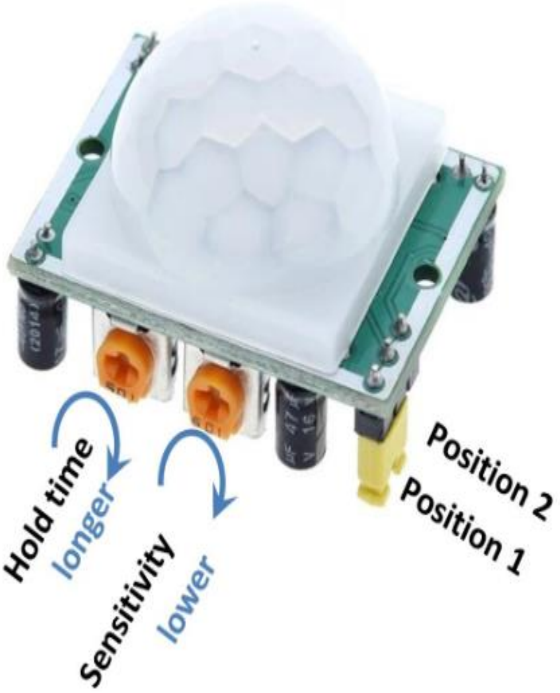
Output signal: 3.3V digital output

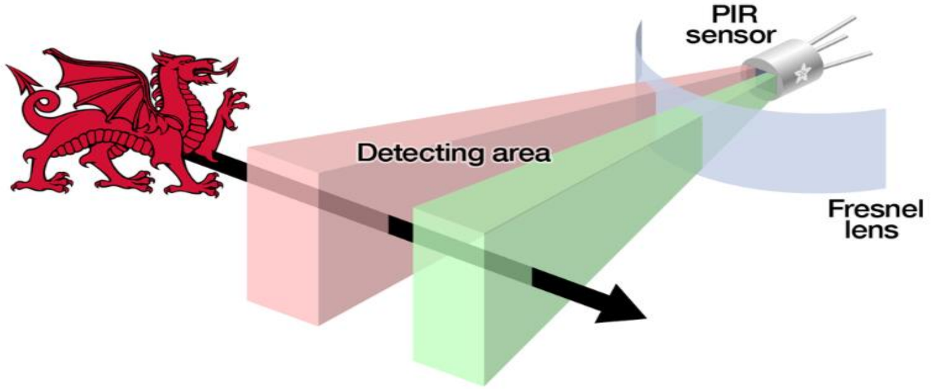
Delay time: adjustable from 0.3 seconds to 5 minutes

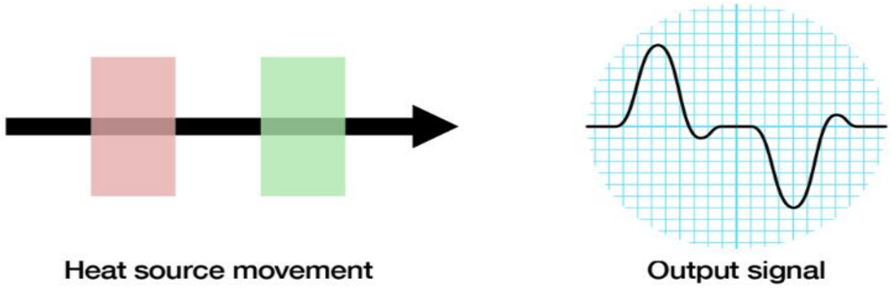
Operating temperature: -15°C to +70°C

Sensitivity: Adjustable


Data
GND
+5..12V


Hold time
longer
Sensitivity
lower
Position 2
Position 1

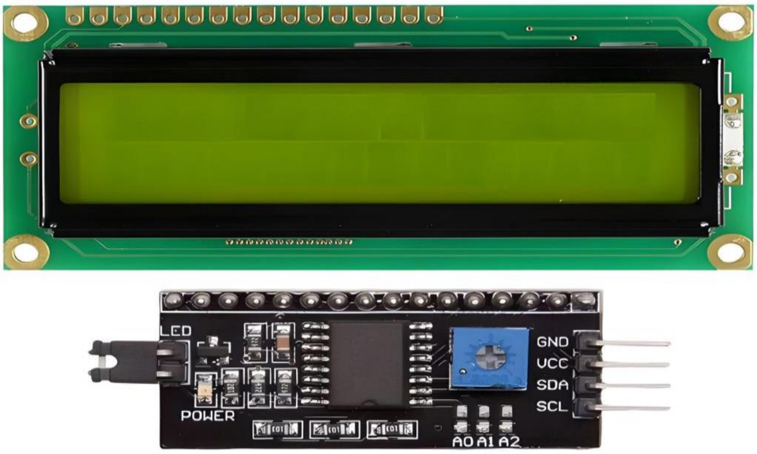

PIR sensor
Detecting area
Fresnel lens


Heat source movement
Output signal

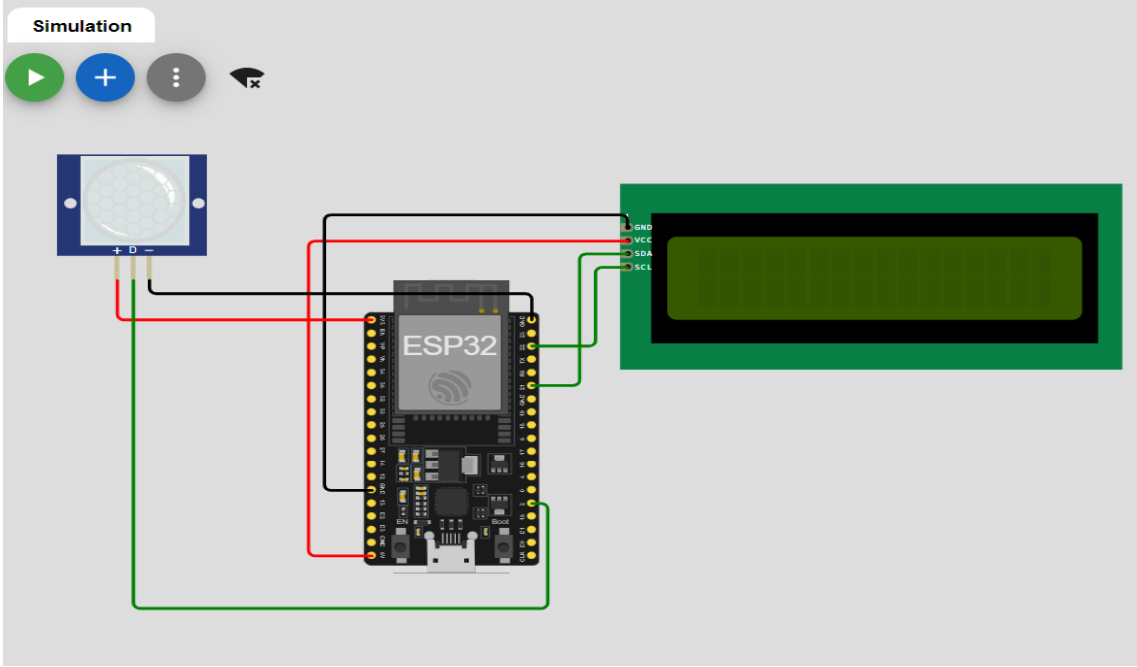
iii) LCD I2C:

The I2C is a type of serial bus , which uses two bidirectional lines, called SDA (Serial Data Line) and SCL (Serial Clock Line). Both must be connected via pulled-up resistors. The usage voltages are standard as 5V and 3.3V. If you already have the I2C adapter soldered onto the board, the wiring is quite easy. You should usually have only four pins to hook up. VCC and GND of course. The LCD display works with 5 Volts. So we go for the 5V Pin.

- **Display capacity:** 16 character x 2 row
- **Display color:** Green backlit
- **Character size:** 2.95 mm wide x 4.35 mm high
- **Character pixels:** 5 W x 7 H
- **Voltage requirements:** 5 VDC +/- 0.5V
- **Current requirements:** 2 mA @ 5 VDC
- **Connection:** 4-pin male header with 0.1″: spacing
- **Communication:** I2C



CIRCUIT DIAGRAM:



SOFTWARE DESCRIPTION:

i) ARDUINO IDE:

The Arduino Integrated Development Environment (IDE) is a powerful and user-friendly software application designed to facilitate the programming and uploading of code to Arduino microcontroller boards. It serves as the primary interface for developing, compiling, and debugging Arduino sketches (programs), making it an essential tool for anyone working with Arduino hardware.

- **Simple Interface:** User-friendly and intuitive design, suitable for beginners and experts.
- **Cross-Platform:** Available for Windows, macOS, and Linux.
- **Board Support:** Compatible with various Arduino boards (Uno, Nano, Mega, etc.).
- **Built-in Libraries:** Extensive libraries for sensors, displays, motors, and more.

- **Serial Monitor:** Tool for real-time communication and debugging via serial data.
- **Sketch Management:** Easily manage, save, and organize Arduino sketches (programs).
- **Library Manager:** Browse, install, and manage external libraries effortlessly.
- **Basic Debugging Tools:** Includes Serial Monitor and error indicators.
- **Easy Compilation & Uploading:** Simple process to compile and upload code to the board.
- **Community Support:** Backed by a large, active community with ample resources.
- **Extensible:** Supports third-party plugins for additional features.
- **Beginner-Friendly:** Ideal for those new to microcontrollers and electronics.
- **Open Source:** Free to use, modify, and share, encouraging innovation.
- **Continuous Updates:** Regular improvements and new features from the Arduino team.
- **Versatile Applications:** Suitable for a wide range of projects, from simple to complex.

ii) WOKWI:

Wokwi is an online simulation platform designed for electronics and embedded systems projects. It allows users to simulate Arduino, ESP32, and other microcontrollers along with various sensors, displays, and components in a virtual environment. Wokwi is widely used by hobbyists, students, and educators to prototype and test their designs without needing physical hardware.

Real-Time Simulation: Test code and hardware interactions without physical components.

Component Library: Access to a wide range of virtual sensors, displays, and more.

Arduino & ESP32 Support: Write and debug code directly in the browser.

No Installation: Completely web-based, ready to use immediately.

Interactive Learning: Ideal for students and educators to learn and teach electronics.

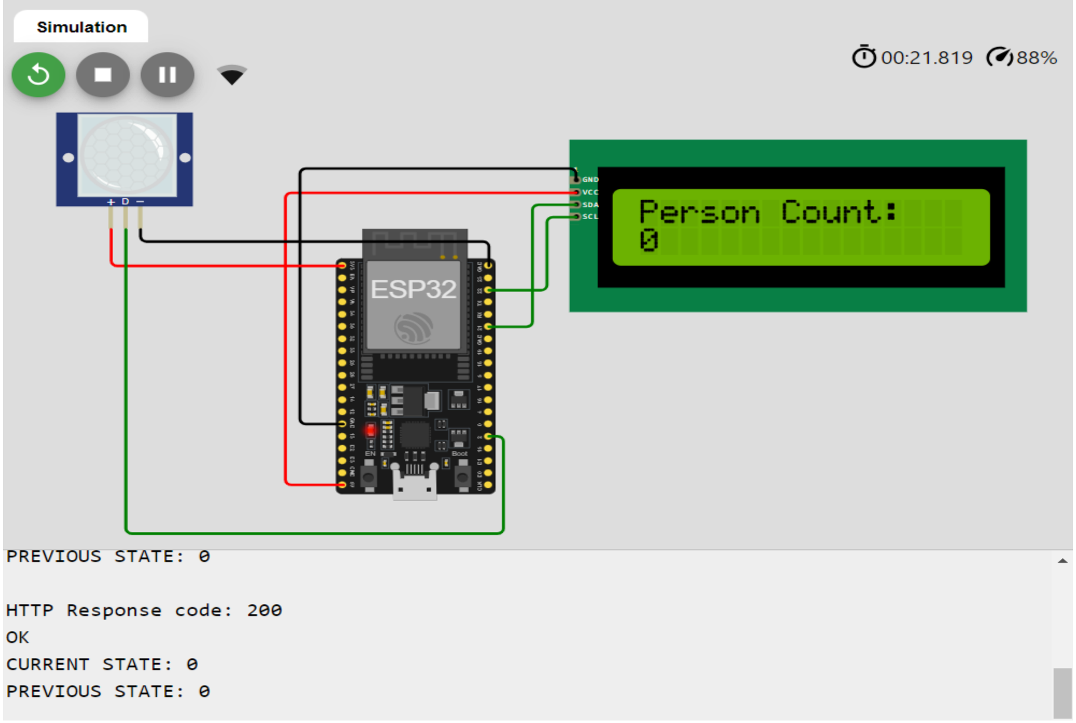
Collaboration: Easily share and collaborate on projects via simple links.

CODE:

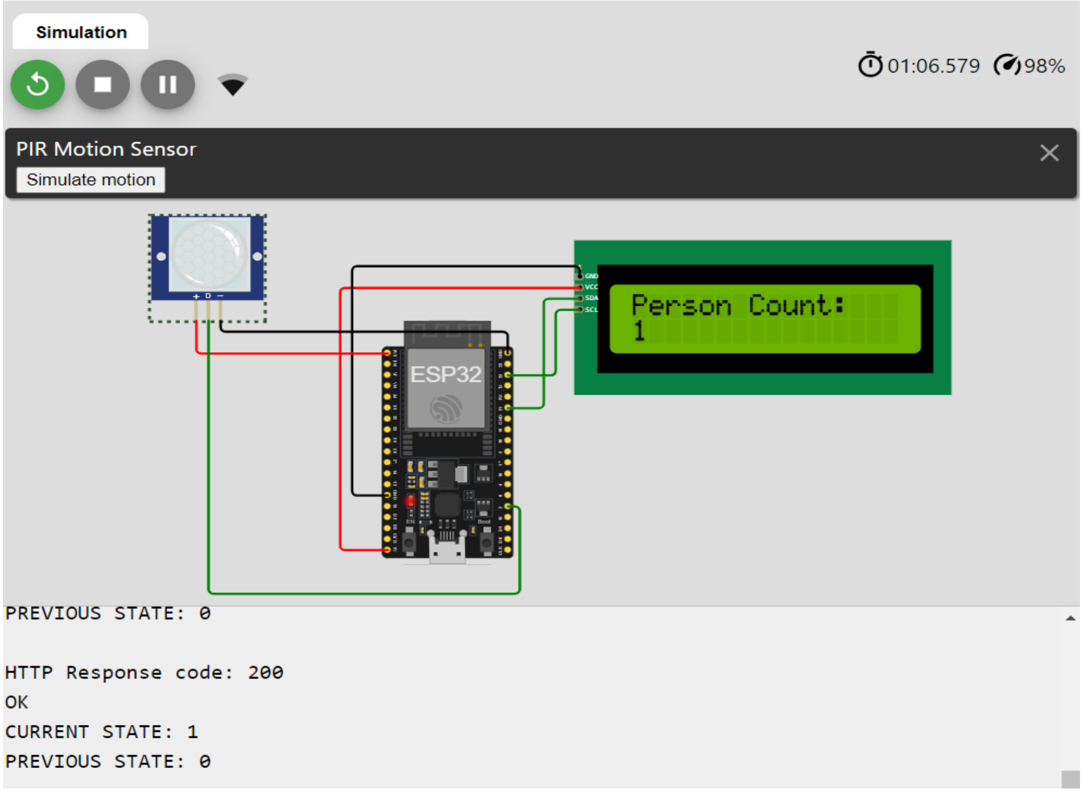
```
#include <WiFi.h>
#include <HTTPClient.h>
#include <LiquidCrystal_I2C.h>
// Initialize the LCD with I2C address 0x27, 16 columns, and 2 rows
LiquidCrystal_I2C lcd(0x27, 16, 2);
// PIR sensor pin
const int pirPin = 2;
// Variables for person count
int personCount = 0;
int previousState = LOW;
int currentState = LOW;
// Wi-Fi credentials
#define WIFI_SSID "Wokwi-GUEST"
#define WIFI_PASSWORD ""
// Server URL and Authorization Token
const char *serverUrl = "https://console.thingzmate.com/api/v1/device-types/7376221501/devices/7376221501/uplink"; // Replace with your server endpoint
String AuthorizationToken = "Bearer fbc003281c7580e80b95006c063d2826";
```

```
void setup() {
  // Initialize the LCD
  lcd.init();
  lcd.backlight();
  Serial.begin(115200);
  // Set the PIR pin as input
  pinMode(pirPin, INPUT);
  // Connect to Wi-Fi
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("Connected to WiFi");
}
void loop() {
  // Read the state of the PIR sensor
  currentState = digitalRead(pirPin);
  Serial.print("CURRENT STATE: ");
  Serial.println(currentState);
  Serial.print("PREVIOUS STATE: ");
Serial.println(previousState);
  Serial.println("");
  // Check if motion is detected
  if (currentState == HIGH && previousState == LOW) {
    personCount++; // Increment person count
    delay(500); // Debounce delay to avoid counting the same person multiple times
  }
  // Update previous state
  previousState = currentState;
  // Clear the display and show person count
  lcd.clear();
  lcd.setCursor(0, 0); // Move cursor to (0, 0)
  lcd.print("Person Count:");
  lcd.setCursor(0, 1); // Move cursor to (0, 1)
  lcd.print(personCount); // Print person count
  // Send person count to the cloud server
  HTTPClient http;
  http.begin(serverUrl);
  http.addHeader("Content-Type", "application/json");
  http.addHeader("Authorization", AuthorizationToken); // Authorization token
  // Create JSON payload
  String payload = "{\"person_count\":" + String(personCount) + "}";
  // Send POST request
  int httpResponseCode = http.POST(payload);
  if (httpResponseCode > 0) {
    String response = http.getString();
    Serial.println("HTTP Response code: " + String(httpResponseCode));
    Serial.println(response);
  } else {
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
  }
  http.end(); // Free resources
  delay(500); // Update the display and send data every second
}
```

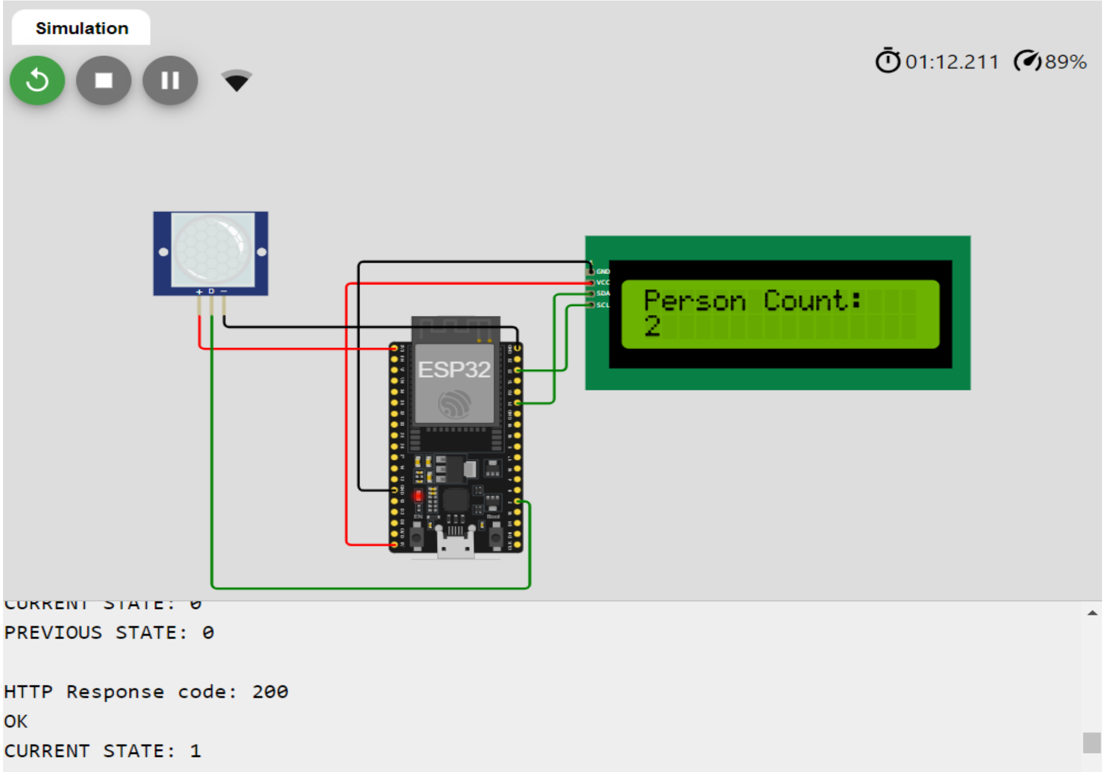
OUTPUT:



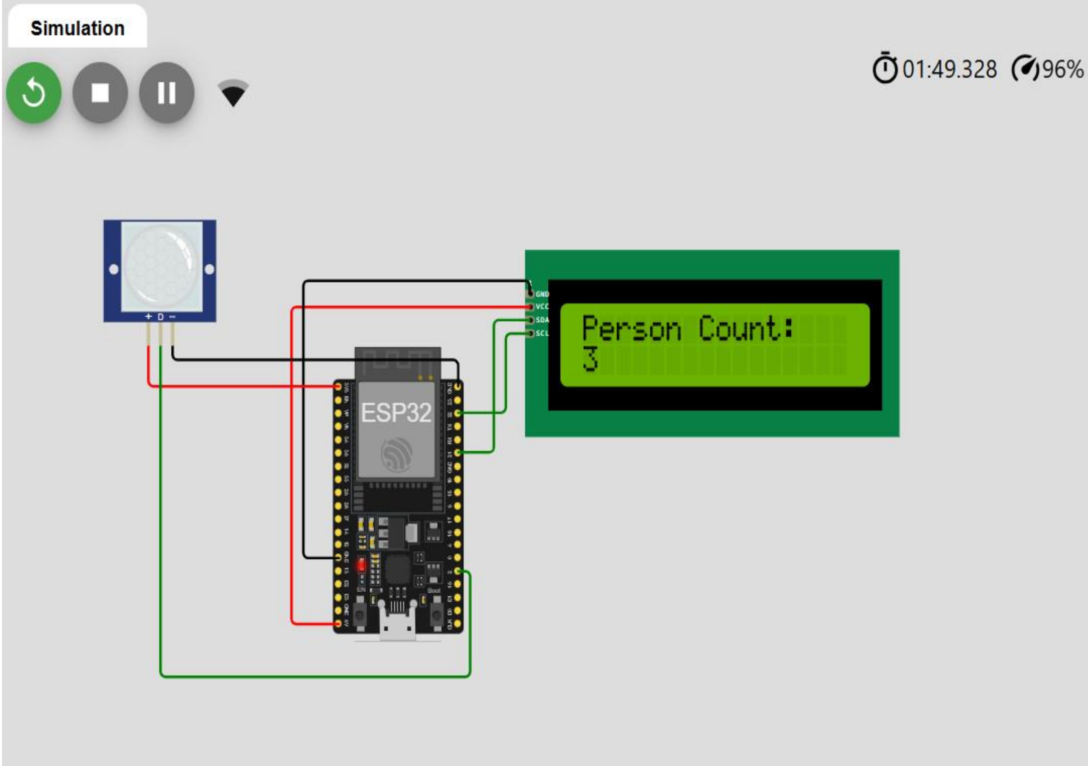
```
PREVIOUS STATE: 0

HTTP Response code: 200
OK
CURRENT STATE: 0
PREVIOUS STATE: 0
```



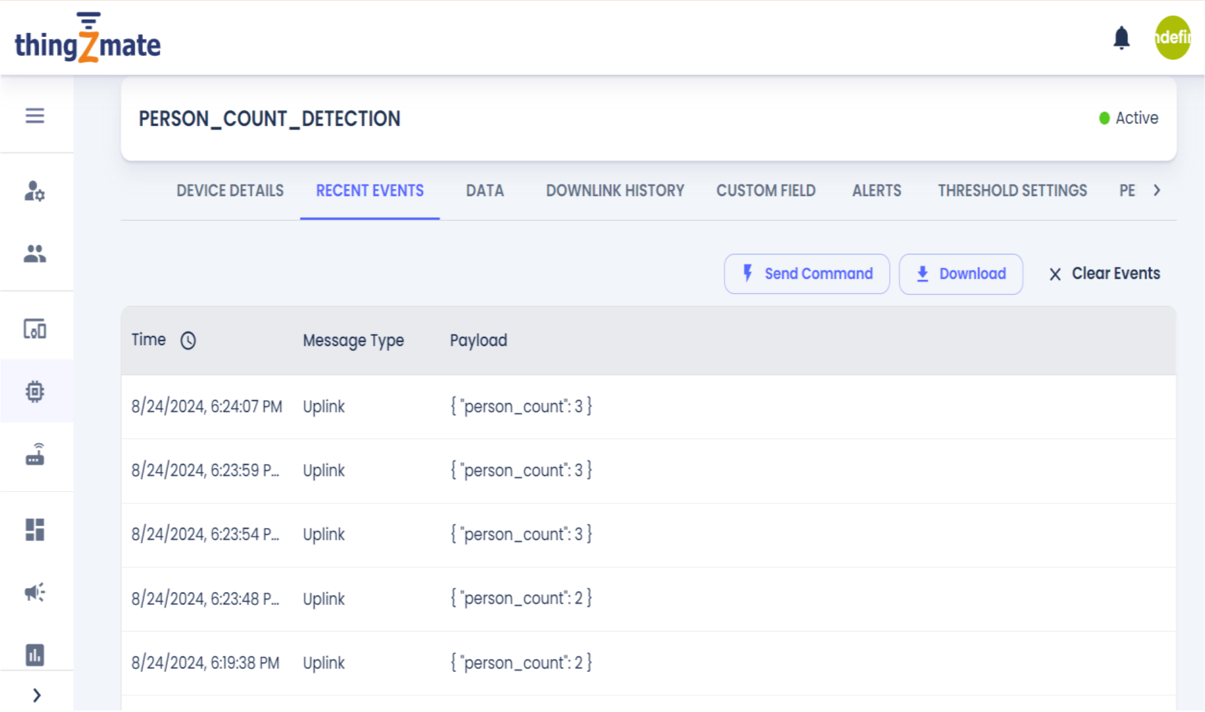
```
PREVIOUS STATE: 0

HTTP Response code: 200
OK
CURRENT STATE: 1
PREVIOUS STATE: 0
```



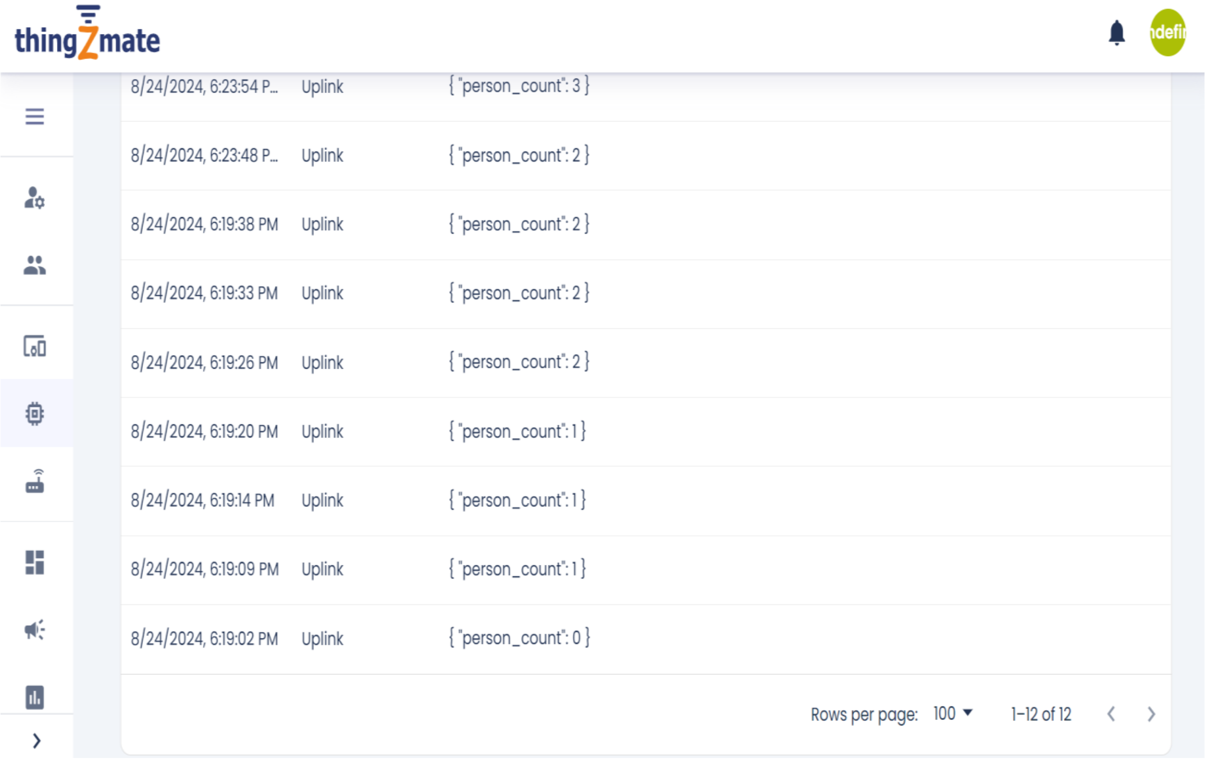
```
CURRENT STATE: 0
PREVIOUS STATE: 0

HTTP Response code: 200
OK
CURRENT STATE: 1
```



CLOUD OUTPUT:

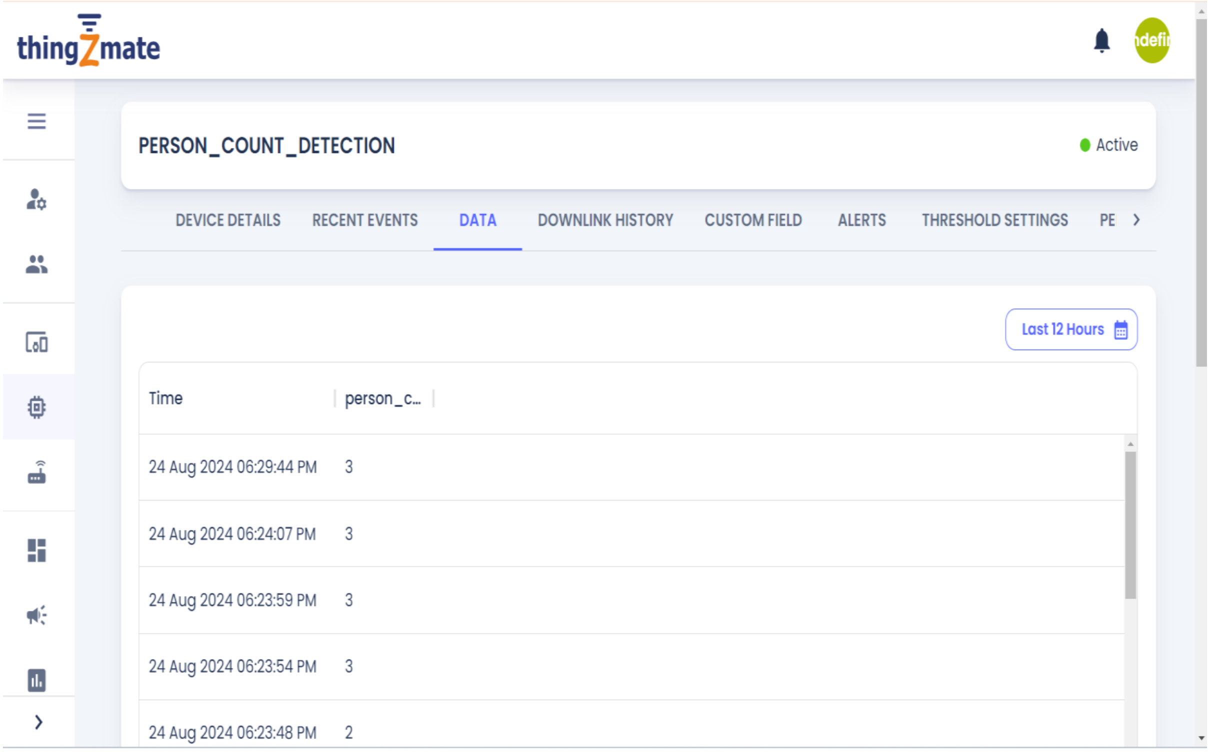


Time	Message Type	Payload
8/24/2024, 6:24:07 PM	Uplink	{ "person_count": 3 }
8/24/2024, 6:23:59 P...	Uplink	{ "person_count": 3 }
8/24/2024, 6:23:54 P...	Uplink	{ "person_count": 3 }
8/24/2024, 6:23:48 P...	Uplink	{ "person_count": 2 }
8/24/2024, 6:19:38 PM	Uplink	{ "person_count": 2 }



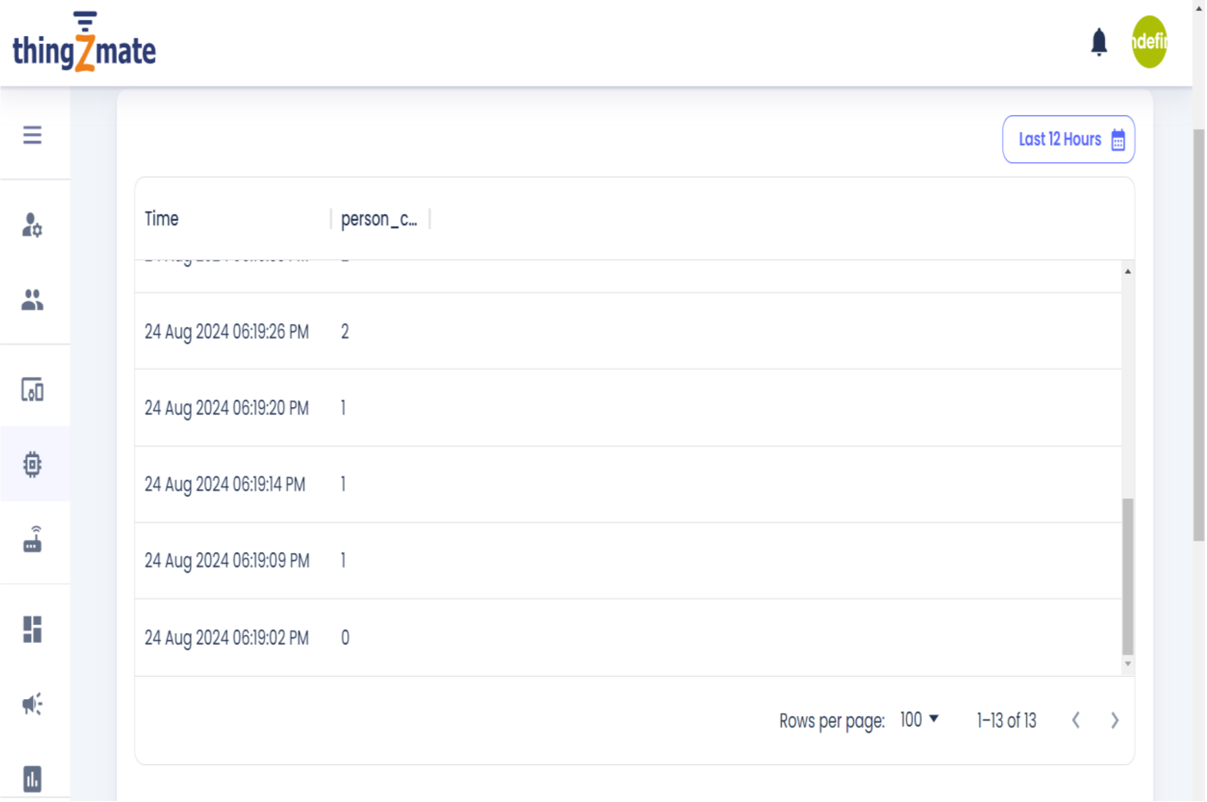
8/24/2024, 6:23:54 P...	Uplink	{ "person_count": 3 }
8/24/2024, 6:23:48 P...	Uplink	{ "person_count": 2 }
8/24/2024, 6:19:38 PM	Uplink	{ "person_count": 2 }
8/24/2024, 6:19:33 PM	Uplink	{ "person_count": 2 }
8/24/2024, 6:19:26 PM	Uplink	{ "person_count": 2 }
8/24/2024, 6:19:20 PM	Uplink	{ "person_count": 1 }
8/24/2024, 6:19:14 PM	Uplink	{ "person_count": 1 }
8/24/2024, 6:19:09 PM	Uplink	{ "person_count": 1 }
8/24/2024, 6:19:02 PM	Uplink	{ "person_count": 0 }

Rows per page: 100 ▾ 1–12 of 12



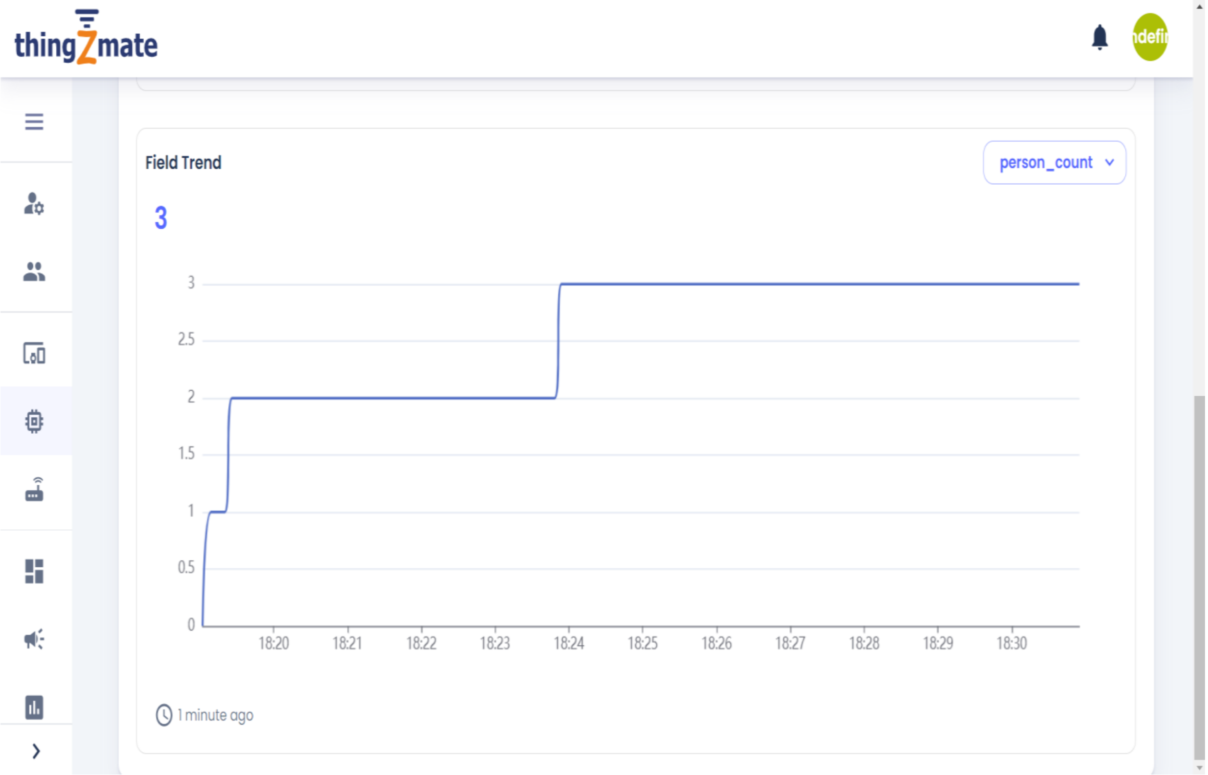
PERSON_COUNT_DETECTION

Time	person_c...
24 Aug 2024 06:29:44 PM	3
24 Aug 2024 06:24:07 PM	3
24 Aug 2024 06:23:59 PM	3
24 Aug 2024 06:23:54 PM	3
24 Aug 2024 06:23:48 PM	2



Time	person_c...
24 Aug 2024 06:19:26 PM	2
24 Aug 2024 06:19:20 PM	1
24 Aug 2024 06:19:14 PM	1
24 Aug 2024 06:19:09 PM	1
24 Aug 2024 06:19:02 PM	0

Rows per page: 100 ▾ 1–13 of 13

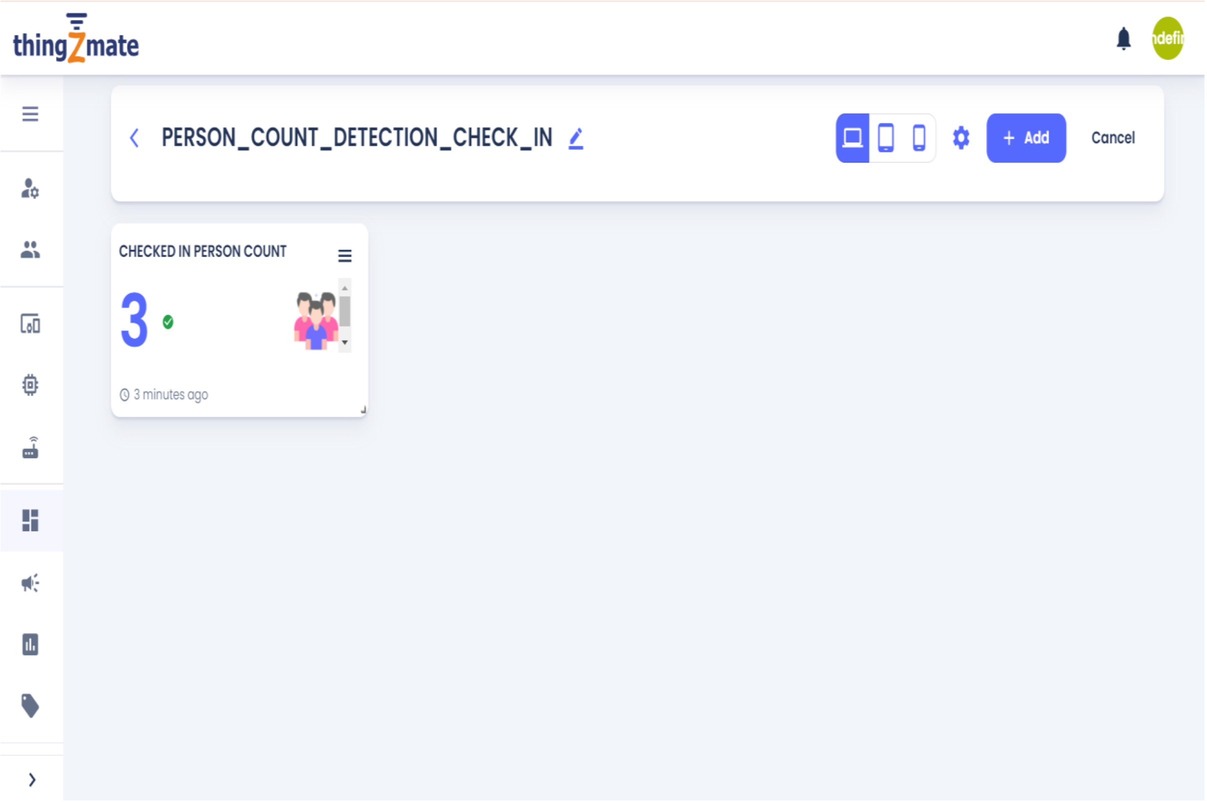


Field Trend

person_count

3

1 minute ago



PERSON_COUNT_DETECTION_CHECK_IN

CHECKED IN PERSON COUNT

3

3 minutes ago

CONCLUSION:

The "Person Count Detection Check-In" system successfully demonstrates the integration of an ESP32 microcontroller, a PIR sensor, and a 16x2 LCD to create an efficient and reliable method for monitoring the number of individuals in a given area. The project's design effectively addresses the need for accurate and real-time person count tracking, providing a practical solution for various applications including space management, safety monitoring, and resource allocation. The system's ability to detect motion and display the count on an LCD provides valuable insights into space utilization and occupancy, which can be crucial for optimizing operations in diverse settings. The use of the ESP32 microcontroller ensures robust performance and flexibility, while the PIR sensor offers reliable motion detection capabilities. Overall, the project highlights the potential of combining accessible electronic components to build functional and impactful solutions. Future enhancements could include integrating additional features such as remote monitoring or data logging to further extend the system's capabilities. The "Person Count Detection Check-In" system represents a significant step towards automating and improving person count management in various environments, contributing to better operational efficiency and safety.